

Invasiv und selbstorganisierend – Die perfekte Kombination für parallele Algorithmen

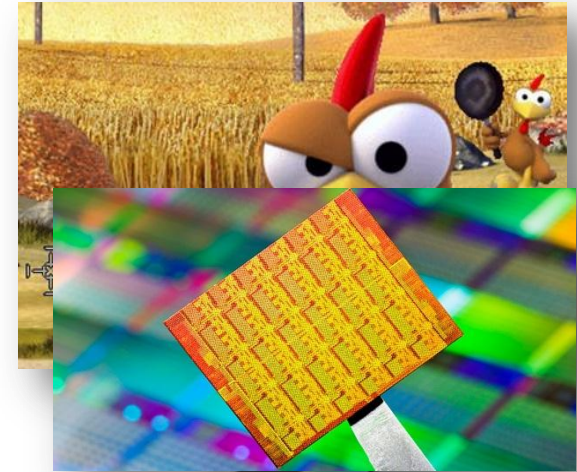
Festsymposium anlässlich des 40-jährigen Jubiläums des Instituts AIFB

Sanaz Mostaghim

INSTITUT AIFB



Von 1971 bis heute ...



Bilder aus Internet



1971

2011/2012

Hochparallele Systeme

Anwendungen in der Wissenschaft:

- **Simulation**

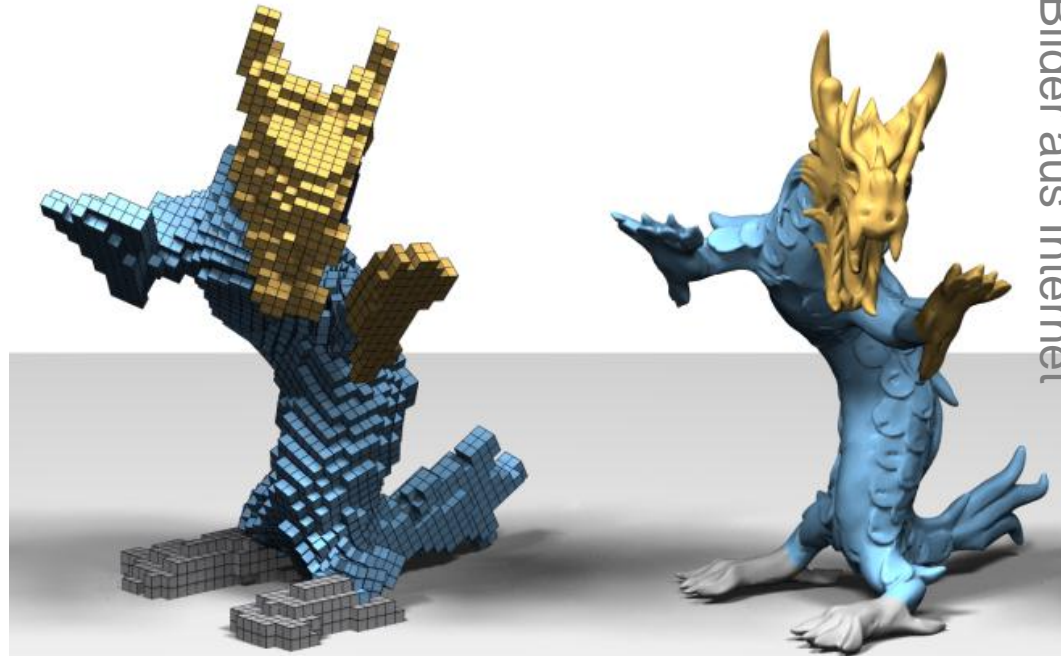
- Klimamodelle, Erdbeben, Tsunamis, Computeranimationen, ...

- **Datenanalyse**

- Medizinische Forschung, Netzwerke, ...

- **Optimierung**

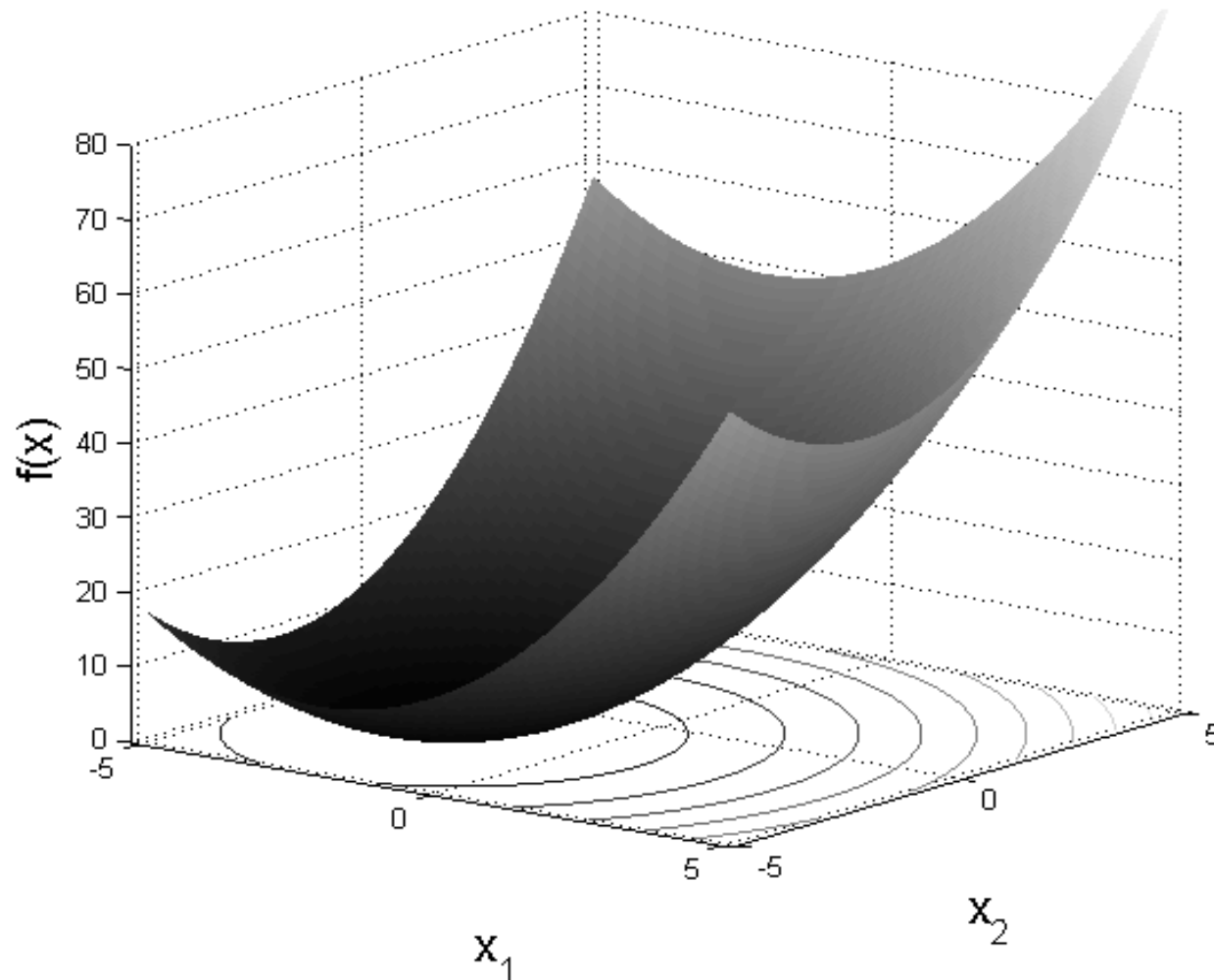
- ...

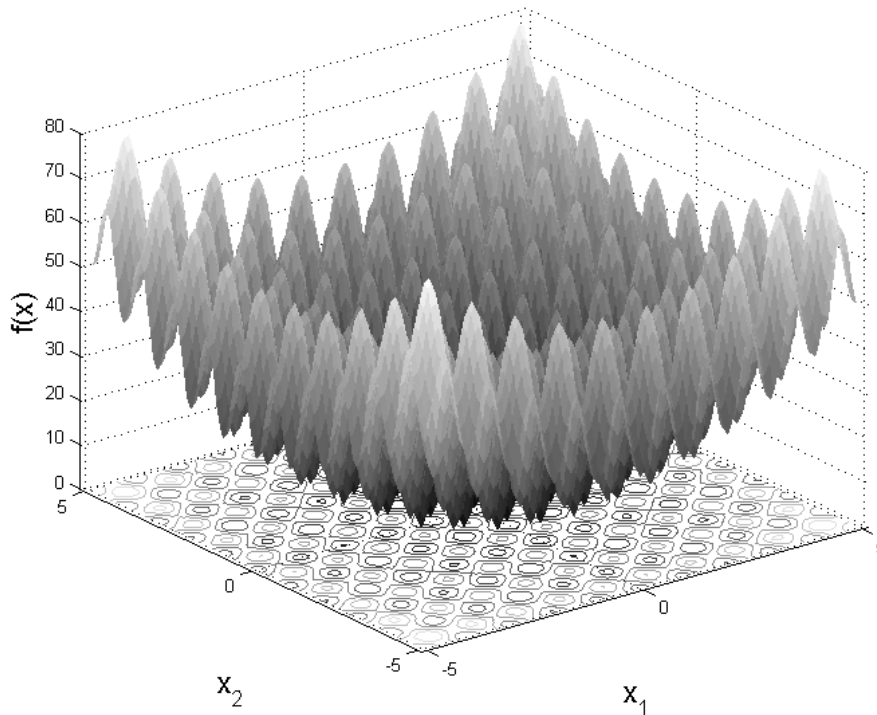


Bilder aus Internet

Optimierung

Conjugate Gradient, 1952





Probleme:

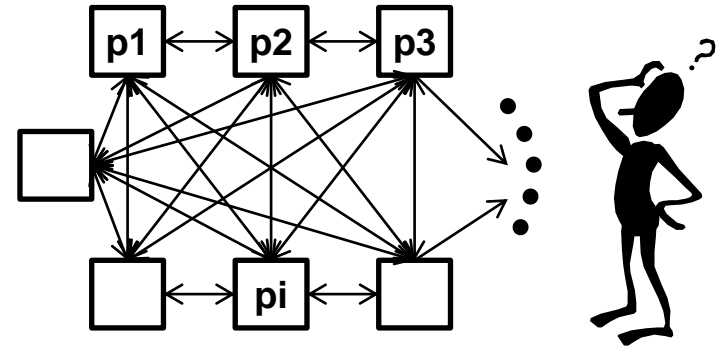
- Nichtlinear
- Hochdimensional
- Nicht differenzierbare Funktionen
- Zeitaufwendige Funktionsevaluationen
- ...

→ Parallele Algorithmen und hochparallele Systeme

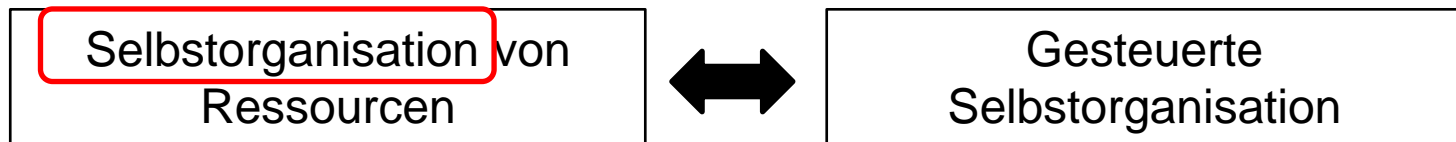
Herausforderungen von hochparallelen Systemen

Komplexität:

- Viele Ressourcen
- Starke Vernetzung
- Datenabhängigkeit
- ...



Dezentrale Lösungsansätze



Organic Computing

Schwarmintelligenz

- Kollektives Verhalten von dezentralen und selbstorganisierenden Systemen
- Lokale Interaktionen
 - Agent-Agent
 - Agent-Umgebung
- Einfache Regeln
- Emergenz → “intelligentes globales Verhalten“



Schwarmintelligenz und parallele Optimierungsalgorithmen

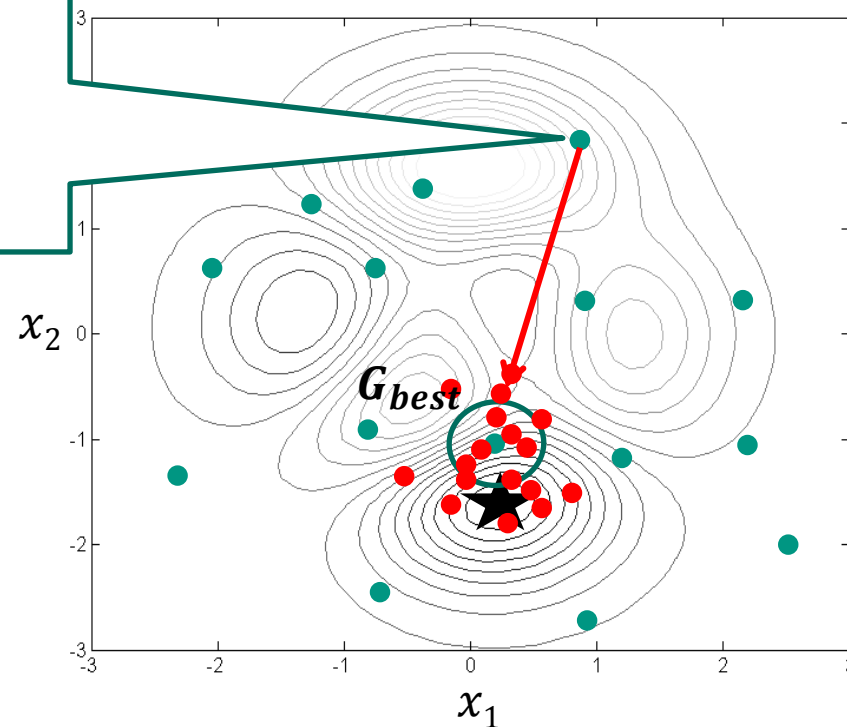
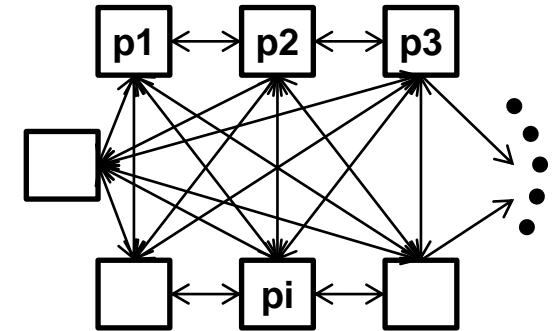
■ Kollektive Suche nach einer optimalen Lösung

Starte mit Lösung $\vec{x}^t \in S$

Wiederhole

1. Finde den Besten in Nachbarschaft: G_{best}
2. Aktualisiere Lösung
 $\vec{x}^{t+1} = \vec{x}^t + \vec{v}^{t+1}$

$$v^{t+1} = wv^t + \varphi_1(P_{best} - x^t) + \varphi_2(G_{best} - x^t)$$

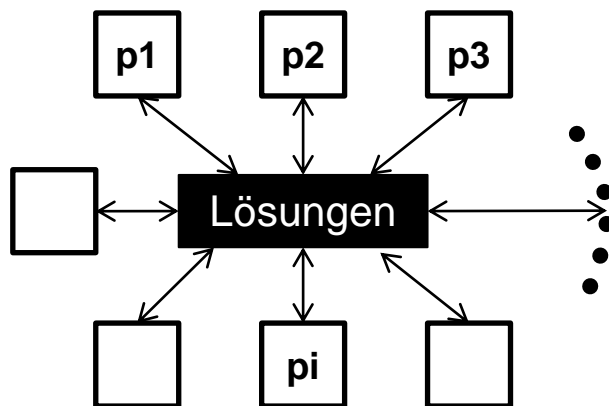


Kollektive Suche nach mehreren Lösungen

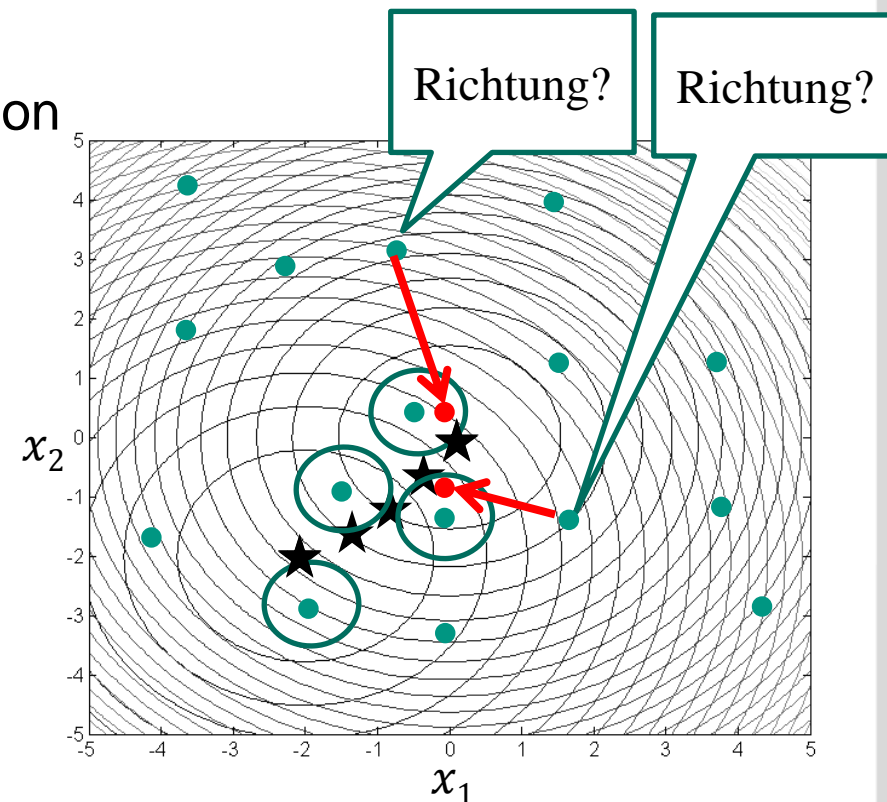
- Optimierte mehrere Funktionen gleichzeitig:

$$\text{Minimiere } \vec{f}(\vec{x}) := (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})) , \text{ s. t. } \vec{x} \in S$$

- Suche nach mehreren Pareto-optimalen Lösungen
- Globales Wissen benötigt
- Aufgabenverteilung durch Kollaboration



Gemeinsamer Speicher



Aufgabenverteilung

■ Paralleler Algorithmus

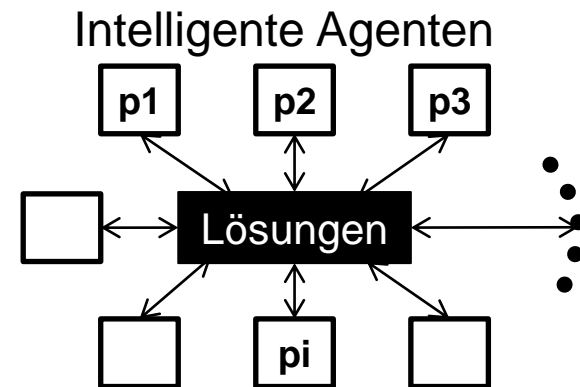
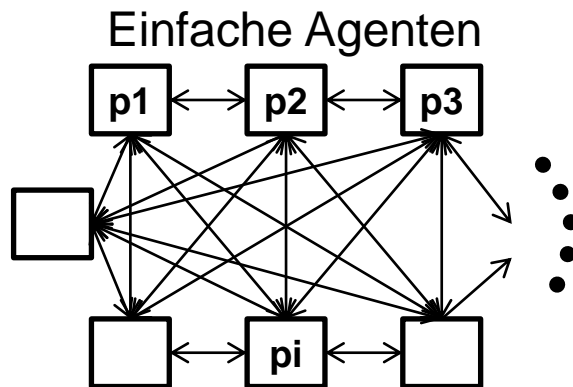
Starte mit Lösung $\vec{x}^t \in S$

Wiederhole

1. Lese gemeinsamen Speicher
2. Wähle Lösung G_{best} , welche Gesamtqualität (Diversität) erhöht
3. Aktualisiere Lösung
 $\vec{x}^{t+1} = \vec{x}^t + \vec{v}^{t+1}$
4. Aktualisiere Lösungen auf dem gemeinsamen Speicher

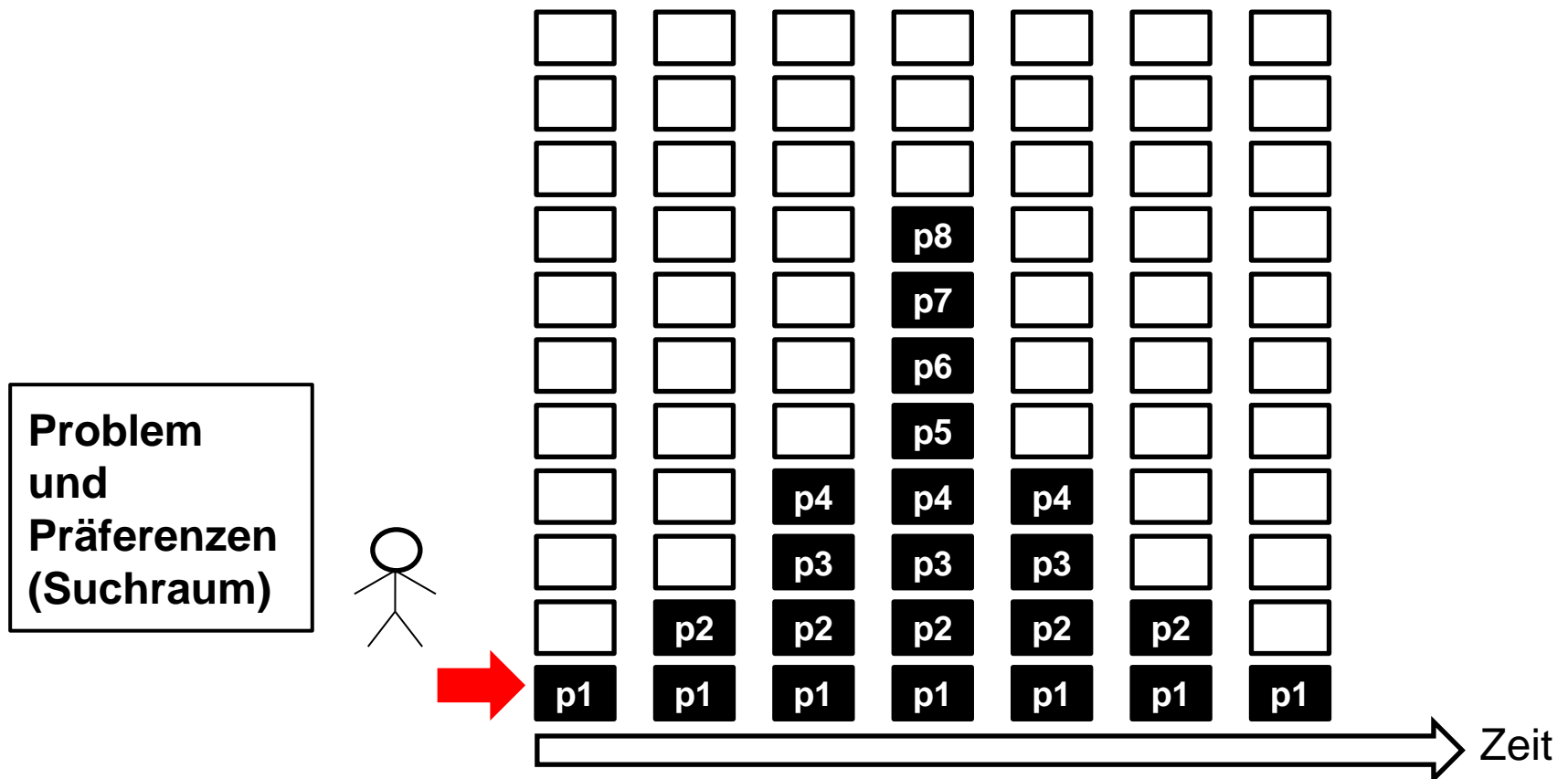
Selbstorganisiert

- Autonome Aufgabenverteilung
- Robust gegen Fehler (Kommunikation oder Ressourcen)
- Geeignet für heterogene Ressourcen
- Skalierbar für beliebige Anzahl von Ressourcen
- ...



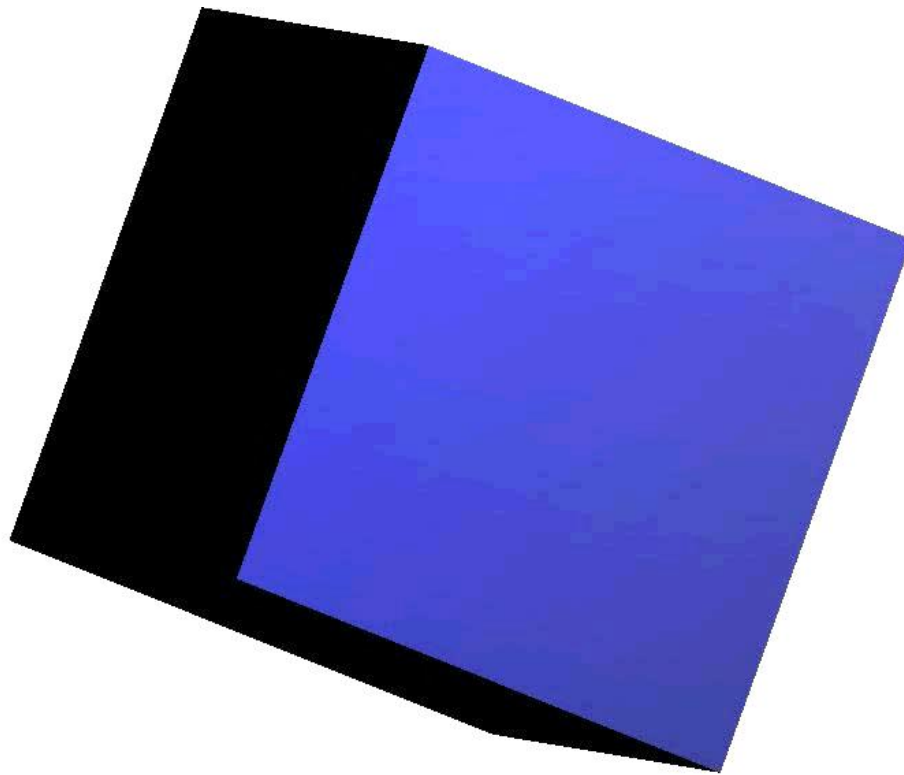
Selbstorganisiert?

- Wie viele Ressourcen?
- Ressourcen entscheiden selbst, nach Bedarf: Invasive Computing [Teich 08]



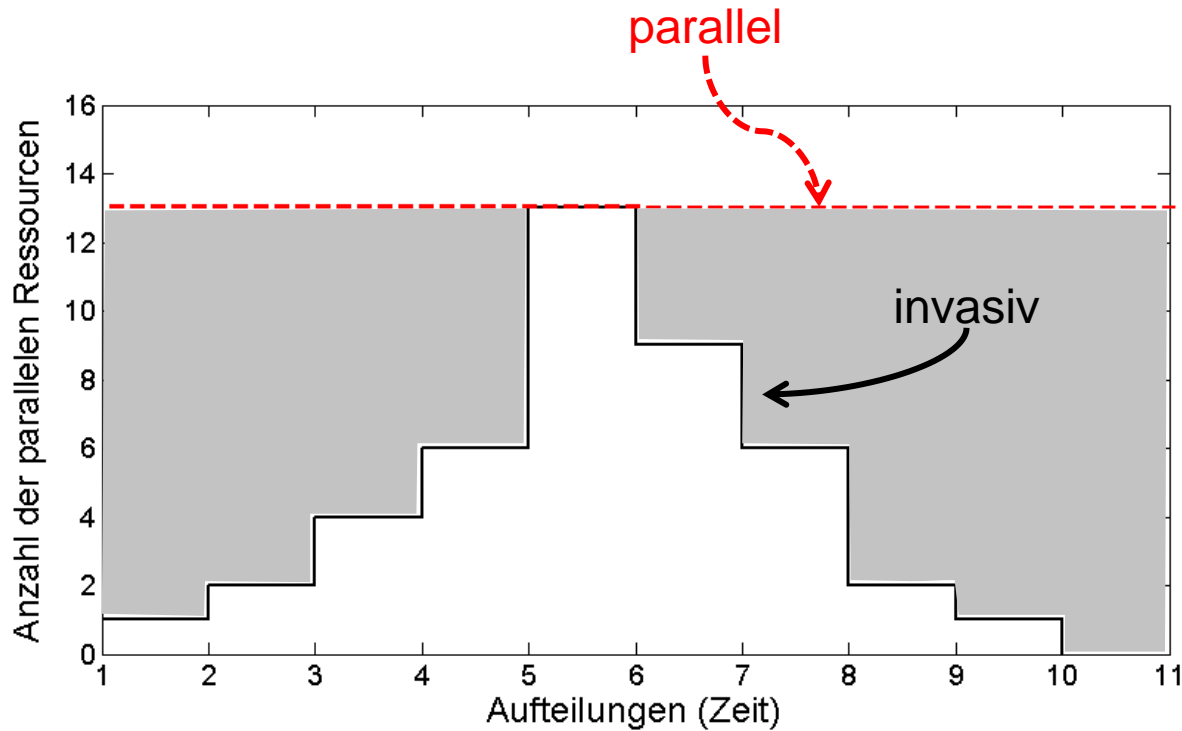
Selbstorganisiert und invasiv

Anzahl der Ressourcen: **640**



Video: [Dellnitz et al.]

Invasiv vs. parallel



Anzahl der Ressourcen	Qualität	Standardabweichung
Testproblem 1		
Invasiv (max. 13)	8,39	0,04
13 parallel	7,45	0,01

[Mostaghim et al. 11a, 11b]

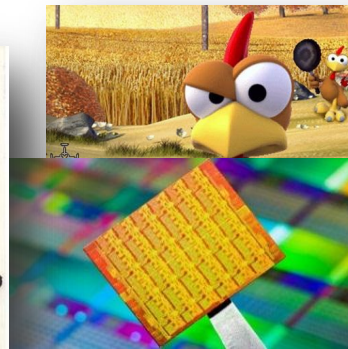
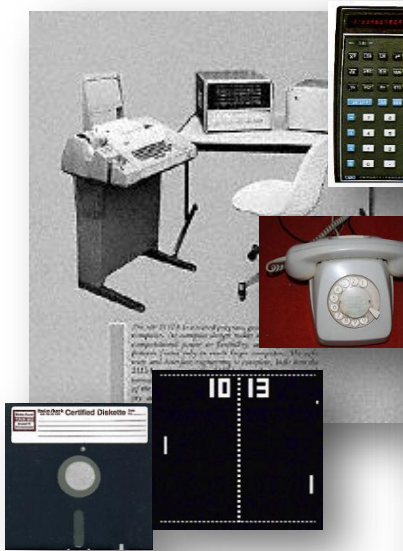
Zusammenfassung

- In der Zukunft
 - Mehr Rechenkapazität in verteilten Ressourcen
 - Bewältigen von heute unlösbaren Problemen aus der Wissenschaft und Industrie

- Dezentrale Ansätze und Selbstorganisation notwendig
 - Organic Computing
 - Invasive Computing

- Invasiv und selbstorganisierend – eine perfekte Kombination für parallele Algorithmen

2051



Bilder aus Internet

1971

2011/2012

2051